Modeling and Optimization of a Multiresolution Remote Image Retrieval System

Antonio Ortega, Zhensheng Zhang

Dept. of Electrical Engineering and CTR
Columbia University, New York, NY 10027

Martin Vetterli

Dept. of EECS, University of California,
Berkeley, CA 94720

ABSTRACT

We study the tradeoffs involved in choosing the bit allocation in a multiresolution remote image retrieval system. Such a system uses a multiresolution image coding scheme so that a user accessing the database will first see a coarse version of the images and will be able to accept or discard a given image faster, without needing to receive all the image data. We formalize the problem of choosing the bit allocation (e.g., in the two resolution case, how many bits should be given to the coarse image and the additional information, respectively?) so that the overall delay in the query is minimized. We provide analytical methods to find the optimal solution under different configurations and show how a good choice of the bit allocation results in a reduction of the overall delay in the query.

1 INTRODUCTION

Consider a generic multiresolution (MR) remote image retrieval system (see [1] for an example of such a system). Users accessing the system will be searching for one or more images within those available in the remote database. The two main components of the system are an image database and a user interface which handles the communication resources transparently to the user. We assume that there are two main stages in a query: (i) the *database search* stage, where in response to the user specification the database manager defines a set of possible candidate images, and (ii) the *browsing* stage, where the user tries to select one or more candidate images, called *target images*. In the latter stage the user is presented with a set of low resolution images (e.g. icons), and can then view them at increasing resolutions, up to the highest available quality, and this until one or more images are selected or the query is terminated. The motivation is that by having fast access first to "coarse" versions of the images, users are allowed to discard, if desired, some of the images *without necessarily having to receive the full quality image*, thus reducing the overall transmission costs of the system. When favoring an MR approach, the underlying assumption is that *the communication costs are the limiting factor*. This situation arises either because (i) the users have access to low-speed (or shared) links, so that transmission delay dominates the total delay in the query (over, for instance, the delay introduced by the search within the database) or simply because (ii) the system has to be designed to minimize the total transmission cost, which we assume to be proportional to the transmission time.

In this work we will concentrate on the browsing stage of the queries. We will further assume that browsing and database search are independent so that our optimization of the browsing stage will not affect the performance of the database search stage. While work reported in the literature has focused on the progressive image transmission

0-8194-1481-6/94/$6.00

schemes [2,3] here we look at the image coding scheme from a systems perspective. Images in the database are coded with an MR scheme (which we do not specify) so that, taking the two-resolution case as an example, at the start of the browsing stage a fraction αB, $0 < \alpha < 1$, of the B bits of the image is transmitted and a low resolution image is reconstructed using those bits. The remaining $(1-\alpha)B$ needed to reconstruct the full resolution image will only be sent if the user requests it. We tackle the problem of assigning a number of bits to each of the image layers (i.e. in our example choosing α) so that the performance of the image retrieval system is optimized.

Note that in a typical bit allocation problem for an MR image coder [4] the objective is to assign bits to each of the image layers to maximize the full quality and possibly to meet some intermediate quality objectives. However here our concern is to study how the bit allocation among the successive image layers affects the overall system performance. As an example, in [1] arbitrary compression rates are chosen for the different resolutions: we point out that this choice can be made so that the system performance is optimized.

To clarify the scope of our optimization, let us note that we can divide the resources used in an MR image retrieval system into roughly three groups: (i) the database computation resources, (ii) the communication resources, and (iii) the computation (including memory) resources at the user sites. We will only consider the latter two resources, under the assumption that the bit allocation only affects the browsing stage and not the search within the database. We can thus state the problem we are seeking to solve as follows, imposing the constraint that all the images in the database use the same allocation:

PROBLEM 1. *How do we allocate the bits to each of the image layers to minimize the total transmission delay or, equivalently, the transmission cost, during a query for a target image.*

At the beginning of the browsing stage, the user is provided with a set of icons from which to select the target image. Since the icons will have very low resolution, it will typically be hard to determine whether the icon set contains a target image and thus the user will have to retrieve some of the images at increasing resolutions in order to make a choice. The trade-off that arises in choosing the bit allocation is clear. If the intermediate resolution were of very high quality (thus requiring a large number of bits, or α close to 1), the user would be able to make a decision on whether the image is acceptable but the cost of retrieving non-acceptable images would be high. Conversely, if the intermediate quality were low (and thus the required bit rate were small, or α close to 0) a decision on the image would not be easy while the cost for choosing a "wrong" icon would be small. The aim of this work is to analyze the trade-off.

This paper is organized as follows. Section 2 provides a more detailed description of a multiresolution image retrieval system and formally defines the parameters of the system as well as our objective function. Section 3 provides solutions to the problem under the different sets of parameters. In particular, it is shown how a "dynamic" queueing–based approach and a "static" average analysis yield the same results. Section 4 draws conclusions and points out areas for further work.

2 SYSTEM DEFINITION

2.1 Multiresolution browsing

Consider the following flow diagram for the user interaction (see Fig. 1). Each user first generates a request and, after a database search, a set of low resolution candidate icons is displayed at the terminal. The user then, at *Stage 1*, selects one of the icons so that its corresponding low resolution image is displayed on the terminal. At *Stage 2*, if (a) the quality of the low resolution image is too poor to decide or (b) the image seems to be adequate for the user requirement, the user requests that the additional information (necessary to create the full resolution picture) is sent (go to *Stage 3*). Otherwise, if the displayed image has sufficient quality and is not one of the targets, it is rejected and another icon is selected (go back to *Stage 1*). At *Stage 3* the full resolution image is displayed and the user can accept it (and terminate the query) or reject it and select another icon (go back to *Stage 1*). The process repeats until an appropriate image is found.

Note that the above description presents a somewhat simplified user interaction since only one candidate image can be considered at any given time. A more general case would not have such a restriction and users would be allowed to store images at different resolutions and then make their decision by comparing those selected. The search can be seen as a process where the user accumulates images at different resolutions (from icon up to full resolution) until the target (one or several images) has been found. While a system with memory might seem

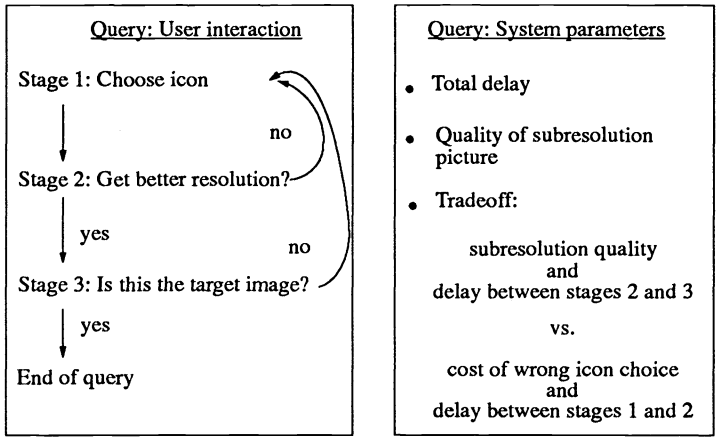

Figure 1: Multiresolution image retrieval system: typical user interaction and corresponding system parameters.

more realistic, our results indicate that, as far as the allocation is concerned, the results are identical in both the memory and memoryless cases.

2.2 System Model

The previous system description can be formalized as follows (refer to Fig. 2). Let t be the probability that an image chosen from the set of icons is one of the *target* images. Let α denote the percentage of the image data volume in the low resolution; we assume that all images are coded using the same parameter α. Let $P(\alpha)$ denote the probability that the quality of the image reconstructed using α percent of the bits is sufficient to make a correct decision (see Section 2.3). Our objective is to obtain α_{opt}, the optimal value of α such that the mean response time is minimized, where the response time is defined as the time interval from the time the request is generated until the time the target image is found.

We model the user interaction (refer again to Fig. 2) by assigning probabilities to the transitions between the successive stages of the query as follows. A transition from *Stage 2* to *Stage 1* occurs when the image has sufficient quality but is not a target, with probability

$$P_{2\to 1} = (1-t)\cdot P(\alpha).$$

A transition from *Stage 2* to *Stage 3* occurs if (a) the image has insufficient quality or (b) if a target image has been found, with probability

$$P_{2\to 3} = 1 - P(\alpha) + t\cdot P(\alpha).$$

Finally at *Stage 3*, the query will end if a target image has been found and will go back to *Stage 1* otherwise, so that we have:

$$P_{3\to 1} = \frac{(1-t)\cdot(1-P(\alpha))}{t\cdot P(\alpha)+1-P(\alpha)}, \quad \text{and} \quad P_{3\to e} = \frac{t}{t\cdot P(\alpha)+1-P(\alpha)}.$$

2.3 Probability of sufficient quality

Given a set of N images, $\mathcal{S}$, assume that we allocate to all of them the same α. We propose to model $P(\alpha)$, the probability that an image, picked at random from the set, has "sufficient" quality for the user to make a decision, as follows. To each image from the set $s_i \in S$, we can associate a rate-distortion (R-D) characteristic, where each

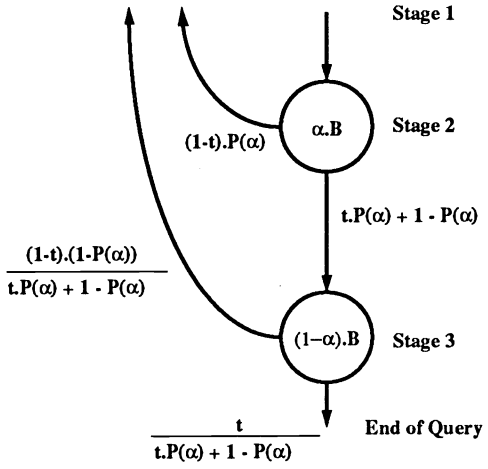

Figure 2: System model for a multiresolution image retrieval system. t is the probability an image is one of the targets. $P(\alpha)$ is the probability that α percent of the total bits provide sufficient quality. B is the image size.

R-D point corresponds to the image coded at one of the available resolutions. Denote these functions as (R_i, D_i) and suppose they are obtained either through measurements on the image set or based upon a model. Note that here we consider any measure of distortion, in particular, measures based on subjective thresholding are possible. We now define a threshold $D_t(s_i)$, which represents the level of "indistinguishable quality", i.e. increasing the rate to reduce the distortion below that threshold produces virtually no improvement to the subjective quality of the image. Then we can define the normalized rate and distortion functions, δ_i and α_i respectively, as:

$$\delta_i = \min(1, D_t(s_i)/D_i), \quad \text{and} \quad \alpha_i = R_i/R_{max}(s_i), \tag{1}$$

where $D_t(s_i)$ can be a common threshold for all images or can be chosen individually for each s_i, and $R_{max}(s_i)$ is the bit rate required by the highest resolution version available for image s_i. Also we define the normalized quality to be $\delta_i = 0$ when no bits are used, i.e. $\alpha_i = 0$.

We can see that $\delta_i(\alpha_i)$ gives an indication of the likelihood that a given image has sufficient quality. For $\delta(\alpha)$ close to one we are close to the full resolution quality so that we will have sufficient quality for almost any application. For $\delta(\alpha)$ close to zero only "easy" searches (e.g. locating a big object within an image) will be possible, others (e.g. locating a texture) will require increased quality. Once we consider $\mathcal{S}$ as a whole, and given that images are picked at random from the set, we can estimate $P(\alpha)$ as follows:

$$P(\alpha) = \frac{1}{N} \sum_{s_i \in \mathcal{S}} \delta_i(\alpha). \tag{2}$$

In the rest of this work we will assume that the probability function $P(\alpha)$ is in the form of $1 - (1 - \alpha)^m$, where m is a positive integer (see Fig. 3). Note that our choice is reasonable when considering typical rate-distortion characteristics and it only affects the exact value of our result; the general analysis holds for more general expressions of $P(\alpha)$.

3 ANALYSIS AND RESULTS

We now provide solutions to the optimization problem outlined in the previous section. Note that we formulated the problem of a single user having access to the database but we now consider the possibility of several users sharing the system. Different methods are called for depending on the exact formulation, in particular

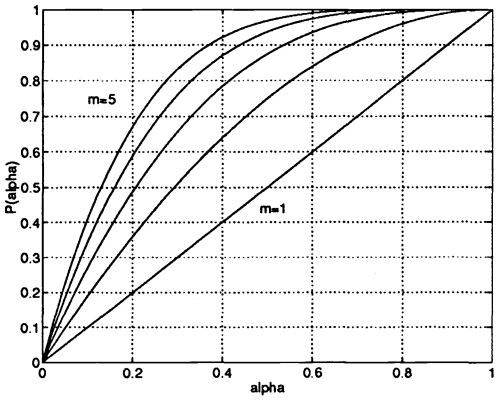

Figure 3: Example of $P(\alpha)$ functions for several values of m. Note that the larger m values are more realistic.

whether the communication resources are shared or not. However, we will show that as far as the optimal point is concerned, most cases of interest yield the same solution. Note also that here we assume $P(\alpha)$ to be the same for all users accessing the database. This does not mean that all users are supposed to access the same set of images, rather it implies that all image sets are similar as far as their quality-rate trade-off. The problem will be tackled first in a queueing framework (Section 3.1) while a static analysis will be proposed later (Section 3.2).

3.1 Dynamic Analysis

3.1.1 Separate channels and large image set

We assume that there are M users in the system who do not share the communications resources and have only to share the computation resources of the database. Idle users, i.e. those not currently browsing through a set of icons, generate a new search request with rate λ. We assume that the database processing time at *Stage 1* is exponentially distributed with mean $1/\mu_1$. This delay is due to the search within the database for the candidate images and the presentation of the icons to the user; the cost reflects the need to transmit the icon set in several batches, given that the user terminal cannot display more than a few icons at a time. The data volume of the images (or equivalently the time it takes to transmit and to display them) is exponentially distributed with mean $1/\mu_2$. For a given value of α, we are interested in the average response time for a user to search for the target image. We model the system as a closed queueing network with four queues, depicted in Fig. 4. The first queue "stores" the users that are currently idle and waiting to generate another query. The remaining queues correspond, respectively, to *Stages 1* to *3* in our model.

Since all the i_1 users at *Stage 1* share the computation resources, the processing rate for one user is μ_1/i_1. The processing rate at which at least one user finishes the process is $i_1(\mu_1/i_1) = \mu_1$. Users at *Stages 2* and *3* have their own communication channels, with transmission rates μ_2/α and $\mu_2/(1-\alpha)$, respectively. If there are i_2 users at *Stage 2* (i_3 users at *Stage 3*, respectively), the rate at which at least one user finishes transmitting is $i_2\mu_2/\alpha$ at *Stage 2* ($i_3\mu_2/(1-\alpha)$ at *Stage 3*, respectively).

In the following, we derive an expression for the mean request delay using the Norton equivalent theorem of queueing networks [5]. The Norton equivalent network with a total of M users is shown in Fig. 5.

To find the state-dependent service rate, s_i, we use the approach presented in [5]. Let

$$p = 1-(1-t)P(\alpha), \quad q = t/(tP(\alpha)+(1-t)(1-P(\alpha))), \quad \mu_1(i) = \mu_1, \quad \mu_2(i) = i\mu_2/\alpha, \quad \mu_3(i) = i\mu_2/(1-\alpha)$$

$$X_i(k) = \prod_{j=1}^{k} \frac{y_i}{\mu_i(j)}, \quad i = 1,2,3, \quad k = 0,1,...,M,$$

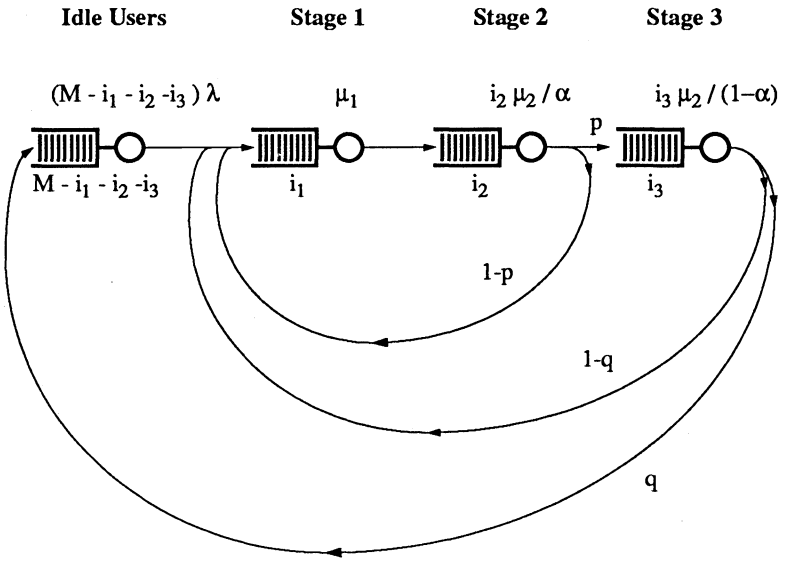

Figure 4: Model for the system as closed queueing network.

where $y_1 = y_2 = 1/pq, y_3 = 1/q, y_4 = 1$ is a solution of the balance equation (equation (4) in [5]).

Let

$$G_1(k) = X_1(k), \quad G_2(k) = \sum_{i=0}^{k} G_1(i)X_2(k-i), \quad k = 0, 1, ..., M, \quad G_3(k) = \sum_{i=0}^{k} G_2(i)X_3(k-i) \tag{3}$$

If we define the state of the system as the number of requests at buffer B in Fig. 5, the state process is a finite population birth-death process with birth-death rates given by

$$\lambda_i = \begin{cases} (M-i)\lambda & 0 \le i \le M-1 \\ 0 & i \ge M \end{cases} \quad \text{and} \quad s_i = \frac{G_3(i-1)}{G_3(i)}, \quad 1 \le i \le M,$$

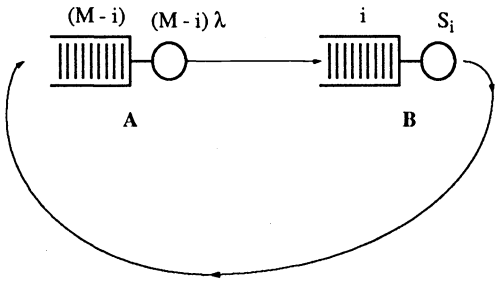

Figure 5: Norton equivalent network.

respectively, where $G_3(i)$ is given in (3) and s_i is the state dependent service rate.

The steady-state mean queue size and request delay can easily be obtained and are given by [6]:

$$E[q] = P_0 \sum_{i=1}^{M} i \prod_{j=0}^{i-1} \frac{\lambda_j}{s_{j+1}} \quad \text{and} \quad E[d] = \frac{E[q]}{\lambda(M - E[q])} \tag{4}$$

respectively, where

$$P_0 = (1 + \sum_{i=1}^{M} \prod_{j=0}^{i-1} \frac{\lambda_j}{s_{j+1}})^{-1}.$$

The optimal value of α is found by minimizing $E[d]$ over α, $0 \leq \alpha \leq 1$. The results are summarized in Figs. 6, 7, 8. The most important point is to note that the optimal operating point is not a function of either t, the number of users M or μ_2. Fig. 6 shows the delay vs. α tradeoff for two values of t. The relative gain of using the α_{opt} is nearly the same in both cases. Fig. 7 shows the same tradeoff for different values of μ_2. Note that in the bottom two curves $\mu_2 \ll \mu_1$ and therefore the delay due to the image transmission dominates the delay due to the database access. However, for $\mu_2 = 0.01$ the dominant term is the database delay and little can be gained by choosing a correct α. As was to be expected, optimizing α only makes sense when communication resources are the bottleneck. In Fig. 8 the service rate for the transmission is only ten times slower than that of the database access and we can see that when the number of users increases over ten the dominating factor becomes the database access delay, and therefore the choice of α does not make as much of a difference (because the users share the database access but not the communication resources).

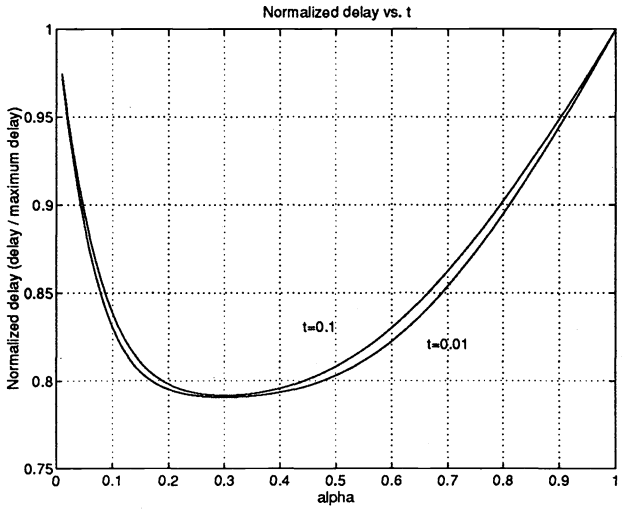

Figure 6: Total delay as a function of α for two values of t. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $M = 10, m = 5, \mu_1 = 0.1, \mu_2 = 0.01, \lambda = 0.1$. Note that the trade-off is practically identical for both values of t. Also, the gain achievable by choosing the optimal operating point would be greater if if μ_2 were much smaller relative to μ_1 since then the delay would be dominated by the image transmission delay.

3.1.2 Separate channels and small image set: non constant t

The results in the previous section indicate that the value of the optimal α does not change with the number of users in the system. In this section, we consider only one user. There are initially N_0 unsearched icons but now we assume that N_0 is "small", so that the probability that one chooses the right icon among i unsearched icons is assumed to be $t(i)$, a function of i. In order to derive an expression for the average delay, $E_\alpha(i)$, incurred

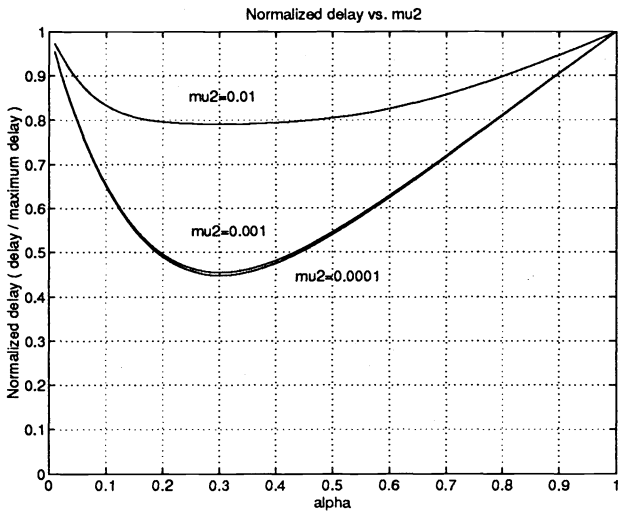

Figure 7: Total delay as a function of α for several values of μ_2. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $M = 10, m = 5, \mu_1 = 0.1, t = 0.05, \lambda = 0.1$. Note that for $\mu_2 = 0.01$ the delay due to the database access, $\mu_1 = 0.1$ is still significant so that optimizing the transmission results in modest gains. Conversely for the other two values of μ_2 transmission dominates the delay and it is clearly advantageous to operate at the optimal point. Also in the latter case, the relative gain is insensitive to the exact value of μ_2, i.e. to the average image size.

in searching the target image given that there are i unsearched icons, renewal theory can be used and it can be shown that (refer to [7] for the details)

$$E_\alpha(i) = (\frac{1}{\mu_1} + \frac{1}{\mu_2})(1 + \sum_{j=2}^{i} \prod_{k=j}^{i} (1 - t(k))) - \frac{1}{\mu_2} P(\alpha)(1-\alpha) \sum_{j=2}^{i} \prod_{k=j}^{i} (1 - t(k)). \quad (5)$$

Since $t(i) \leq 1$ for all i, $1 \leq i \leq N_0$, we have $\sum_{j=2}^{i} \prod_{k=j}^{i} (1 - t(k)) \geq 0$. Therefore, the problem of minimizing $E_\alpha(i)$ subject to $0 \leq \alpha \leq 1$ is equivalent to the problem of maximizing $P(\alpha)(1-\alpha)$ subject to $0 \leq \alpha \leq 1$.

So that we have

$$\alpha_{opt} = \arg \max_{0 \leq \alpha \leq 1} (P(\alpha)(1-\alpha)). \quad (6)$$

For the same set of parameters, α_{opt} is exactly the same as that obtained in the previous section (although here our analysis only covers the single-user case).

3.1.3 Shared Resources

We now consider the case where all the users share one communication channel. As in Section 3.1.1, the system can be modeled as a closed queueing system (see Fig. 4), with modified transmission rates which can be determined as follows. The number of users at *Stages 2* and *3* in Fig. 4 represents the number of users sharing the transmission link. At any given time, a user can either be in *Stage 2* or *Stage 3*, but cannot be in both at the same time. Therefore, since the link is shared, if there are i and j users at *Stages 2* and *3* respectively, the transmission rates for one user would be $\frac{\mu_2}{(i+j)\alpha}$ and $\frac{\mu_2}{(i+j)(1-\alpha)}$ at *Stages 2* and *3*, respectively. The rate at which at least one user finishes transmitting would be $\frac{i\mu_2}{(i+j)\alpha}$ and $\frac{j\mu_2}{(i+j)(1-\alpha)}$ at *Stages 2* and *3*, respectively.

Because the transmission rate depends on the number of users at other queueing systems, we cannot use the Norton equivalent theorem of queueing networks. Instead, we solve the steady state probability directly. A detailed description of the solution goes beyond the scope of this paper and can be found in [7].

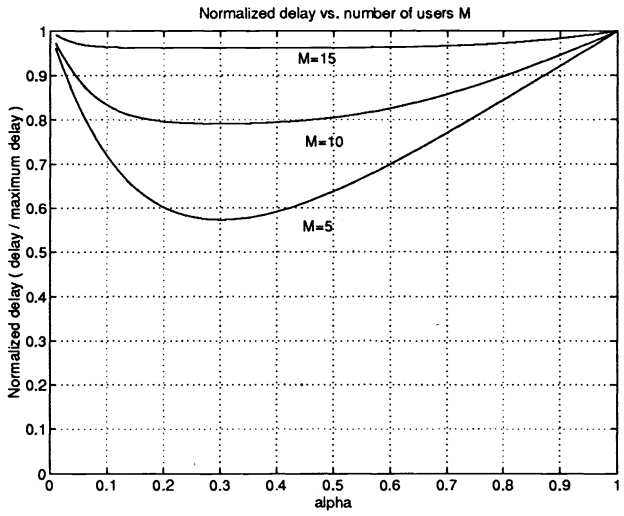

Figure 8: Total delay as a function of α for several values of the number of users. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, t = 0.05, \lambda = 0.1$. Note that, as we maintain μ_1 constant and all users share the database, increases in the number of users imply that the database delay becomes more significant and the gains obtained by optimizing the transmission are smaller.

Numerical results for $M \leq 10$ indicate that the optimal value of α is independent of the value of M and once again identical to that obtained in Section 3. Fig. 9 shows the delay vs. α tradeoff for different number of users, while Fig. 10 shows the tradeoff when t varies.

3.2 Static Analysis

We deemed "dynamic" the analysis introduced above because we set up a model where queries could be terminated at any given time with a certain probability. We now impose the restriction that the user has to examine *all* the candidate images to make a decision. We describe this approach as "static" in the sense that the order in which the images are examined no longer matters, and one can concentrate on the average behavior.

The user can select N_1 icons to be expanded to an intermediate quality level based on the following criterion: if the image quality is sufficient expand the image *only* if it is one of the targets, else expand regardless. Out of the N_1 intermediate resolution images again N_2 are selected to be displayed at full resolution. The same criterion is applied at this step. Note that the main difference between this model and the previous one is that here the user selects images in batches, whereas before a sequential selection was performed. Thus we are in effect modeling the user interaction "with memory" mentioned in Section 2.1. Suppose that we know there are n images in the original set of icons that can be of interest to the user, then at every step at least n images are selected while in addition some images are selected because quality was not sufficient to decide. The user will make a final choice among the N_2 remaining images but since all of them have already been downloaded we assume there is no cost involved and we will ignore this step in our model.

Assume that, if B is the number of bits for the original images, then αB bits are used for the intermediate resolution images (where $0 \leq \alpha \leq 1$). Even though the images have different sizes, in order to make the selection the user will have to go through all of them so that the optimal cost depends only on the relative sizes of the subresolution images, i.e. α, and not on the actual sizes of the images. Therefore, to simplify our analysis we can assume all images have the same size. Note that this is consistent with the results of the dynamic analysis where the α_{opt} did not change with μ_2. Our aim is to minimize the total transmission cost:

$$J = N_2 B + (N_1 - N_2)\alpha B \tag{7}$$

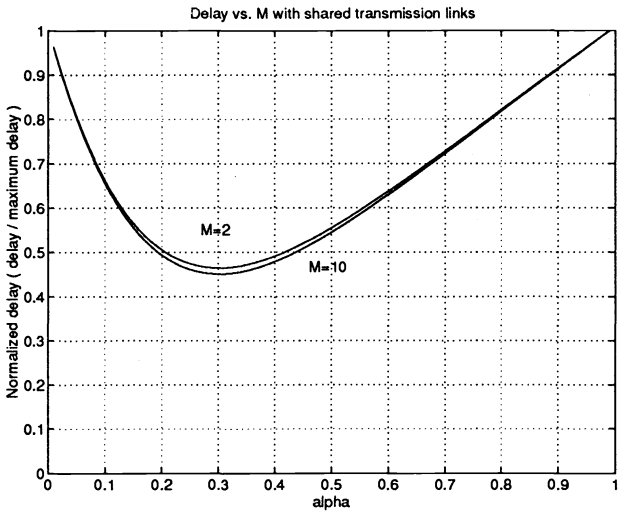

Figure 9: Total delay as a function of α for two values of the number of users when the commnunication resources are shared. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, t = 0.05, \lambda = 0.1$. Note that the relative gain is practically the same regardless of the number of users. This is due to the fact that all users access a set of images with the same characteristics.

m	α
2	0.4227
3	0.3700
4	0.3313
5	0.3012

Table 1: Optimal α for several expressions of $P(\alpha)$. Solutions obtained using the static analysis.

where we just have added the bit rates of the images at each resolution level, or, equivalently,

$$J' = N_2 + (N_1 - N_2)\alpha \; = \; (1-\alpha)N_2 + \alpha N_1. \tag{8}$$

Now assuming the number of images is sufficiently large, we have that on average:

$$N_2 = P(\alpha)n + (1 - P(\alpha))N_1. \tag{9}$$

Using (9) in (8) we get:

$$J' = N_1 + (n - N_1)P(\alpha)(1-\alpha). \tag{10}$$

Therefore, since $n \leq N_1$ our objective in order to minimize the cost is to maximize $P(\alpha)(1-\alpha)$, as we had already seen in Section 3.1.2. This can be done for a given function $P(\cdot)$ and the solutions obtained for $P(\alpha) = 1 - (1-\alpha)^m$ for $m = 2, \cdots, 5$ are summarized in Table 1. The results match those obtained following the dynamic analyses of Section 3.1.

The above analysis can be generalized to the case where the system allows the images to be downloaded at more than two different resolutions. Our results indicate that increasing the number of resolutions in the system decreases the average delay in the query. Fig. 11 shows the delay obtained at the optimal operating point for several cases with more than two layers. A detailed description of the results is given in [7].

4 DISCUSSION AND CONCLUSIONS

A first conclusion of the foregoing sections is that finding the optimal operating point can be worthwhile in reducing the overall delay, in particular in cases where users are connected to the database through low-speed links. For instance choosing the optimal α can provide reductions in delay of up to a factor of two in the $m = 5$ case (see Fig. 7 for example).

A second point is to note that the optimal α is independent of the exact procedure that is used for browsing, as shown by the identical α_{opt}'s obtained in Sections 3.1 and 3.2. Similarly we find the same results for α whether one or several users access the database, and whether or not the users share the transmission resources. Finally, we see no dependence of the optimal result on the size of the initial image set, or the probability of getting the correct image, t (equivalently, in the static analysis, the number of correct images n).

The intuitive justification is that the exact procedure for retrieving the images is not relevant because we are concerned with minimizing an average cost. Since for every image we have an average measure of the "risk" of having to retrieve the rest of the image (i.e. $P(\alpha)$) and we assume all images are identical (i.e. same probability) it is normal to expect that the only factor to determine the optimal operating point would be $P(\alpha)$.

Similarly, as we increase the number of users, and even if the transmission resources are shared, the optimal value for α remains unchanged. This is again due to our choosing to minimize the average delay for a set of users that are identical, at least in a statistical sense.

Finally, it should be mentioned that our analysis of the multiple resolution layers case indicates that substantial reductions in delay are possible by using more than one intermediate resolution. Systems with several intermediate resolutions should thus be considered provided that the increase in implementation complexity can be afforded.

To summarize our results, we have (i) proposed a simple model for user interaction in an MR image retrieval system, (ii) shown that it is advantageous in terms of delay and transmission cost to choose the optimal operating point, and (iii) indicated the benefits of using more than two resolution layers.

Our analysis leaves a number of questions for future work. In particular it would be of interest to perform quality measures on real images to obtain empirical expressions for $P(\alpha)$. Also, since the average analysis provides the same results as the dynamic one, it would be interesting to relax the constraint on the bit allocation and allow each image to have a different α. This type of optimization could be achieved within the static analysis framework.

5 REFERENCES

[1] N. D. Degan, R. Lancini, P. Migliorati, and S. Pozzi, "Still images retrieval from a remote database: the system imagine," *Signal Processing: Image Communications*, vol. 5, pp. 219–234, May 1993.

[2] K. Knowlton, "Progressive transmission of grey-scale and binary pictures by simple efficient and lossless encoding schemes," *Proceedings of the IEEE*, vol. 68, pp. 885–896, July 1980.

[3] M. Malak and J. Baker, "An image database for low bandwidth communication links," in *Proc. of the Data Compression Conference*, (Snowbird, Utah), March 1991.

[4] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Proc.*, 1994. To appear.

[5] K. Chandy, U. Herzog, and L. Woo, "Parametric analysis of queueing networks," *IBM J. Res. Develop.*, pp. 36–42, January 1975.

[6] D. Gross and C. Harris, *Fundamentals of Queueuing Theory.* New York: John Wiley, 2nd ed., 1985.

[7] A. Ortega, Z. Zhang, and M. Vetterli, "Modeling and optimization of a multiresolution image remote retrieval system," tech. rep., Center for Telecomm. Research, Columbia University, 1994.

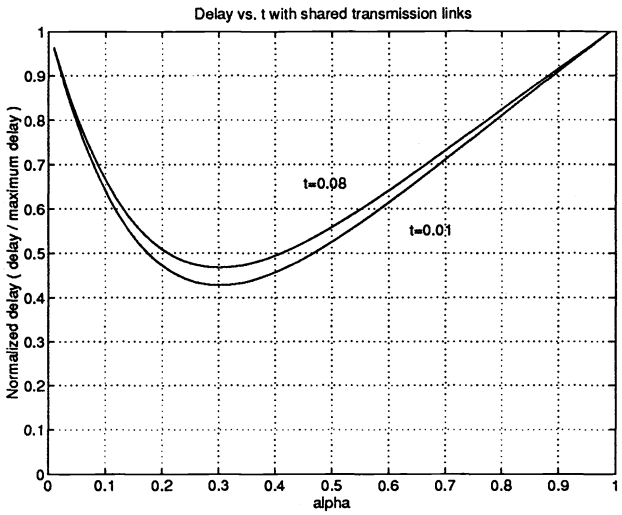

Figure 10: Total delay as a function of α for two values of t when the communication resources are shared. In all cases we have that $\alpha_{opt} = 0.3012$. The other parameters are set to $m = 5, \mu_1 = 0.1, \mu_2 = 0.01, M = 5, \lambda = 0.1$

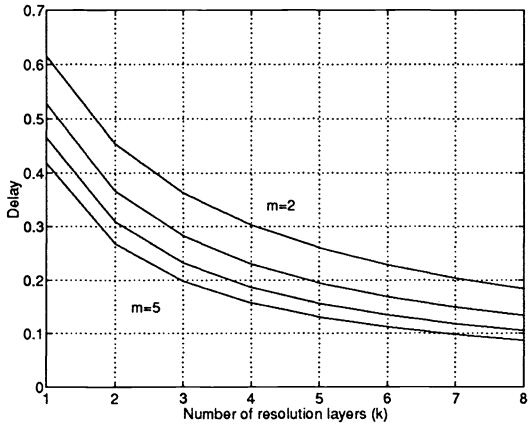

Figure 11: Overall delay at the optimal combination of α's for different values of m. Note how the delay can be significantly decreased by increasing the number of resolutions, at the cost of increased complexity.